## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A computer implemented method of generating a spoken dialogue application, comprising:

generating a two-dimensional graphical representation of a call flow which does not alter finite state machines in real time;

generating a context free grammar representation of the call flow using [[said]] the two-dimensional graphical representation;

generating a finite state machine from the context free grammar representation of the call flow, the finite state machine comprising a plurality of nodes including at least a first leaf node and at least a first root node; and

generating a dialogue application code for a spoken dialogue application from said finite state machine, wherein said generated dialogue application code for said functions are executable during runtime of said spoken dialogue application for walking the finite state machine from the at least one root to the at least one leaf of the finite state machine.

2. (Cancelled)

3. (Currently Amended) The method of claim 1, wherein [[said]] the two-dimensional graphical representation is generated using standardized graphical elements.

4. (Currently Amended) The method of claim 1, wherein [[said]] the two-dimensional graphical representation is generated using VISIO.

5. (Original) The method of claim 1, wherein said context free grammar representation is in a Backus-Naur Form format.

6. (Original) The method of claim 5, wherein said context free grammar representation is in augmented Backus-Naur Form format.

7. (Original) The method of claim 1, wherein a function is associated with a node in said finite state machine.

8. (Original) The method of claim 1, further comprising customizing generated application code.

9. (Original) The method of claim 1, wherein generated application code associated with an output function performs a table lookup prompt information.

10. (Currently Amended) A computer-readable medium that stores instructions for controlling a computer device to generate a spoken dialog application, the instructions comprising:

generating a two-dimensional graphical representation of a call flow which does not alter finite state machines in real time;

generating a context free grammar representation of the call flow using the two-dimensional graphical representation;

generating a finite state machine from the context free grammar representation of a call flow, the finite state machine comprising a plurality of nodes including at least a first leaf node and at least a first root node; and

generating a dialogue application code for a spoken dialogue application from said finite state machine, wherein said generated application dialogue code for said functions are executable during runtime of said spoken dialog application for walking the finite state machine from the at least one root to the at least one leaf of the finite state machine.

11. (Currently Amended) A system for generating a spoken dialog application comprising:

a processor in communication with a module, wherein the module is configured to generate a finite state machine from a context free grammar representation of a call flow generated using a two-

dimensional graphical representation of the call flow, wherein generating the two-dimensional graphical representation does not alter finite state machines in real time, wherein the finite state machine comprises a plurality of nodes including at least a first leaf node and at least a first root node ; and

wherein the module is configured to generate application code using said finite state machine, wherein the application dialogue code is generated dependent on how said finite state machine is traversed, for functions to be executed upon state transitions in said generated finite state machine, wherein said generated application code for said functions are executable during runtime of said spoken dialog application wherein the finite state machine is traversed from the at least one root to the at least one leaf of the finite state machine.

12. (Currently Amended) A spoken dialog application method, comprising:

traversing a finite state machine, that is generated from a context free grammar representation of a call flow generated using a two-dimensional graphical representation of the call flow, wherein generating the two-dimensional graphical representation does not alter finite state machines in real time, and comprises at least a first root node and at least a first leaf node;

generating application code as said finite state machine is traversed from the at least one root to the at least one leaf of the finite state machine [[,]] ; and

invoking said generated application code for functions associated with nodes in said finite state machine, wherein each node of said finite state machine is mapped to a corresponding function.

13. (Original) The method of claim 12, wherein said context free grammar representation is generated from a graphical representation of said call flow.

14. (Original) The method of claim 12, wherein said context free grammar representation is in a Backus-Naur Form format.

15. (Original) The method of claim 14, wherein said context free grammar representation is in an augmented Backus-Naur Form format.

16. (Original) The method of claim 12, wherein generated application code performs a table lookup for prompt information.

17. (Currently Amended) A spoken dialog system, comprising:

means for traversing a finite state machine that is generated from a context free grammar representation of a call flow generated using a two-dimensional graphical representation of the call flow, wherein generating the two-dimensional graphical representation does not alter finite state machines in real time;

means for generating application code as the finite state machine is traversed using the finite state machine; and

means for invoking said application code for functions associated with nodes in said finite state machine, wherein each node of said finite state machine is mapped to a corresponding function.